

| Name | | Kennzahl | Matrikelnummer |
|--------------|--|--|----------------|
| | | | |
| | | Bitte tragen Sie sofort mit Kugelschreiber <i>Name, Kennzahl</i> und <i>Matrikelnummer</i> ein! Halten Sie bitte den Studentenausweis bereit! Lesen Sie die Angaben genau, bevor Sie beginnen! Verwenden Sie bitte nur diesen Bogen für Ihre Antworten! Die Prüfungsdauer beträgt 90 Minuten. | |
| Summe | | | |
| 1.Bsp. | | Institut für Computersprachen (Prof. Brockhaus) | |
| 2.Bsp. | | | |
| 3.Bsp. | | | |
| 4.Bsp. | | | |

Ich trete zum ... mal zur Prüfung an.

Prüfung aus Übersetzerbau 21.6.2001

1. 25 % Quadrupel-Code

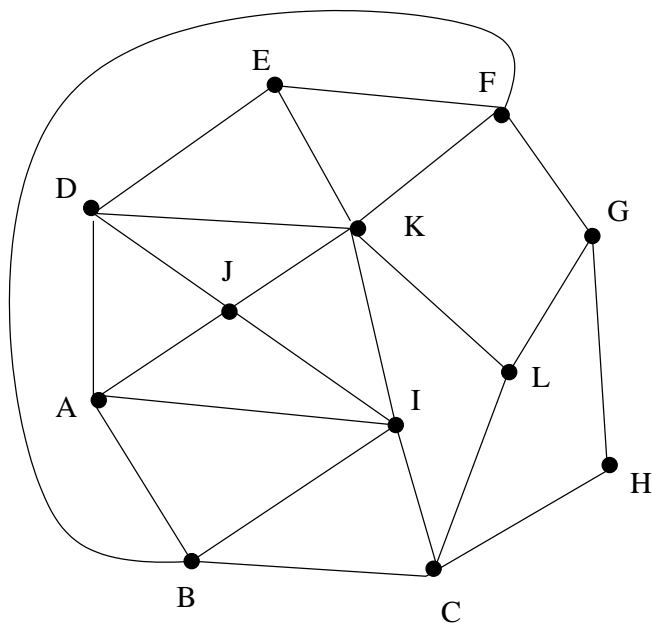
Erzeugen Sie für das rechte Programmstück Quadrupel-Code nach der Kontrollflußmethode. Ein INTEGER ist 4 Byte und ein LONG 8 Byte groß. Die Untergrenze aller Indexbereiche ist 0.

```

VAR
  ia: ARRAY[10,5] OF INTEGER;
  la: ARRAY[15] OF LONG;
  m,n: INTEGER;
  l: LONG
:
IF NOT( (n>4) ) OR (m<>n) ) THEN
  l := la[ia[m,n]];
ELSE
  l := la[m+5];
END
  
```

2. 25 % Konfliktgraph und Auslagerungskosten

Gegeben sind der Konfliktgraph und die Auslagerungskosten.



| R | Kosten |
|---|--------|
| A | 12 |
| B | 3 |
| C | 1 |
| D | 12 |
| E | 30 |
| F | 30 |
| G | 1 |
| H | 1 |
| I | 12 |
| J | 8 |
| K | 15 |
| L | 1 |

Bestimmen Sie die *Abarbeitungsreihenfolge* und den Grad der Knoten und geben Sie für *jedes auszulagernde* Pseudoregister den Wert der Prioritätsfunktion $P(n)$ an. Nehmen Sie an, dass **drei** reale Register zur Verfügung stehen.

| Nr. | Register | Grad | ev. $P(n)$ | Nr. | Register | Grad | ev. $P(n)$ |
|-----|----------|------|------------|-----|----------|------|------------|
| 1. | | | | 7. | | | |
| 2. | | | | 8. | | | |
| 3. | | | | 9. | | | |
| 4. | | | | 10. | | | |
| 5. | | | | 11. | | | |
| 6. | | | | 12. | | | |

3. 25 % Grammatik

Gegeben sei folgende Grammatik (Kleinbuchstaben und Sonderzeichen sind Terminalsymbole; Startsymbol ist S):

$S \rightarrow L \$$
 $L \rightarrow a$
 $L \rightarrow L L$
 $L \rightarrow b L c$

a) (5 %) Bestimmen Sie die first- und follow-Menge des Nonterminals L .

b) (5 %) Begründen Sie, warum die Grammatik nicht LL(1) ist.

c) (15 %) Geben Sie eine neue Grammatik an, die die selbe Sprache erzeugt, und eine LL(1) Grammatik ist.

4. 25 % Attributierte Grammatik

Ein Kochbuch KB besteht aus einer Liste von Rezepten. Es gibt mindestens ein Rezept. Nach dem Rezeptnamen `rezn` stehen die Zutaten. Jedes Rezept hat mindestens eine Zutat. Eine Zutat Z hat einen Zutatnamen `zun` und die Menge `num` in Gramm, die für dieses Rezept benötigt wird. Die Rezepte sind durch "." abgeschlossen, z.B.:

```
Biskuit Mehl 100 Zucker 100 Ei 100.  
Nockerl Mehl 200 Wasser 50 Salz 2.
```

a) (8 %) Erstellen Sie eine Grammatik, die ein ganzes Kochbuch einlesen kann. (Verwenden Sie dazu die Terminalsymbole "`rezn`", "`zun`", "`num`" und "." und ignorieren Sie etwaige Leerzeichen.)

b) (17 %) Erweitern Sie die Grammatik aus a) um Attribute zur Berechnung des Energiewertes eines Rezepts. Die Ausgabe soll so wie das Eingabeformat sein, nur dass zusätzlich nach einem Rezeptnamen der Energiewert in KJoule steht. Die Ausgabe soll im synthetisierten Attribut "`KB.x`" erfolgen. Im Attribut "`zun.j`" steht der Energiewert für ein Gramm einer Zutat in KJoule, in "`rezn.n`" und "`zun.n`" steht der Name, und in "`num.z`" der Zahlenwert des entsprechenden Terminalsymbols. Der Operator "|" hängt Zeichen- und Zahlen intelligent aneinander ("`Wasser`"|100 ergibt "`Wasser 100`"). (Ignorieren Sie auch hier etwaige Leerzeichen.) Das obige Beispiel ergibt (Mehl.j=1; Zucker.j=2; Ei.j=2; Wasser.j=0; Salz.j=0):

```
Biskuit 500 Mehl 100 Zucker 100 Ei 100.  
Nockerl 200 Mehl 200 Wasser 50 Salz 2.
```